

IBM FileNet Case Analyzer Cubes' Deep Customizations

Part 1: Introduction, business requirements and sample environment

[XYZ Insurance Company as an Example]

[March 14, 2012]

IBM

By Yiwei Song

Abstract: IBM FileNet Case Analyzer, formerly Process Analyzer, provides several cubes and corresponding Excel report templates by default. Reports can be customized by combining different dimensions, measures or calculations, and key data fields can be also appended as dimensions. However, in some customer cases, such customizations are not sufficient to meet specific requirements. This 5-article series introduces several ways of direct cube customizations. This article is the part 1 of series, gives an overall introduction, explains requirements and sets up the sample environment.

Yiwei Song, a member of CDL ECM CoE (Center of Excellence) in China, certified FileNet (4.0) developer. Has 5 years FileNet engagement experience, especially familiar with FileNet BPM. Acted as product expert and served several customer cases across insurance, banking and E&U industries. You may reach him at <http://linkedin.com/in/yiweisong>.

Special Note

This 5-article series introduces 4 ways of cube customizations, which can achieve:

- Organization-user drill up or drill down;
- Last month amount, month-on-month growth rate;
- Same month last year amount, monthly growth rate on a year-over-year basis;
- 72-hour/7-day/30-day/3-month completion rate.

This article is the part 1 of series, gives an overall introduction, explains requirements and sets up the sample environment.

Prerequisites

This article assumes readers are familiar with IBM FileNet BPM 4.5.x or 5.0. Readers should have experience of using FileNet Case Analyzer (or formerly Process Analyzer). Additionally, readers should have basic knowledge of OLAP (i.e. Online Analytical Processing) and LDAP (i.e. Lightweight Directory Access Protocol).

Introduction to IBM FileNet Case Analyzer

IBM FileNet Case Analyzer (CA for short), formerly FileNet Process Analyzer, is a core component of FileNet BPM. The CA provides a mechanism of generating statistic reports from case/process history, which are important for customer to track the load of FileNet BPM based system, govern the organization performance and gain insights from process data.

CA provides various kinds of reports based on process data or case data. This article focuses on those based on process data. Case related reports are not in the scope.

CA provides reports (according to infoCenter

http://publib.boulder.ibm.com/infocenter/p8docs/v5r0m0/topic/com.ibm.p8.pa.user.doc/bp_fpa018.htm) like:

- How much work is currently in a particular queue
- How much new work entered the system during a time period
- How much work was completed during a time period

CA reports can be either Excel Pivot Tables or Cognos BI reports.

Technically, CA depends heavily on OLAP. CA periodically extracts data from PE EventLog, does some ETL (Extraction Transformation Loading) and stores fact data in CA OLAP fact tables, then refreshes CA OLAP cubes. Excel or Cognos BI consumes these OLAP cubes and produces final reports.

Figure 1. Dataflow of Case Analyzer for process events



Reports generated by CA greatly benefit customers; however some customers still have particular requirements beyond existing CA OLAP cubes. OLAP cube customization can meet some typical requirements.

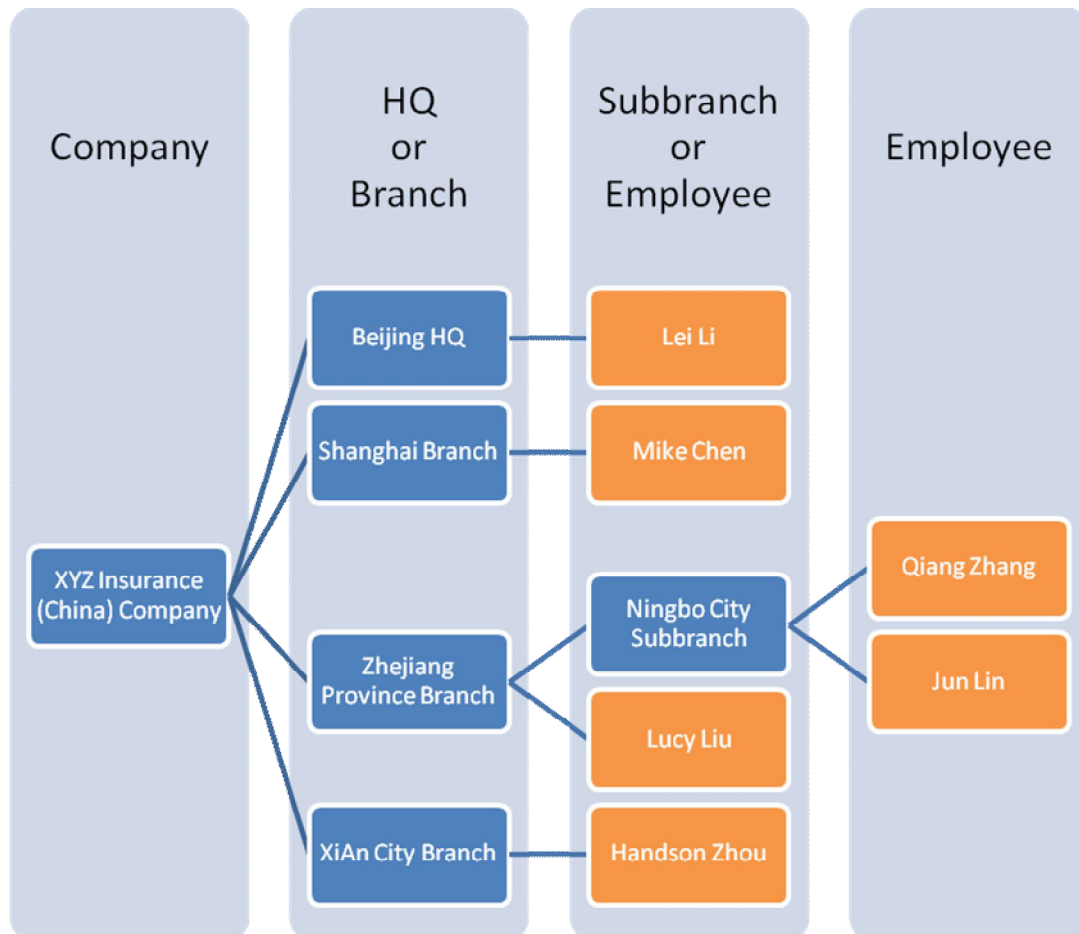
Customizations introduced in this article are applicable to either CA 5.x or PA 4.5.x. This article uses a mock company, XYZ Insurance Company as example, analyzes their requirements, and proposes workable solutions by customizing CA OLAP cubes.

XYZ Insurance Company's requirement on process statistic report

XYZ Insurance Company (XYZ Insurance for short) is an insurance company in China. It runs life insurance business all over the country, so it has many branches or sub-branches in provinces and cities. All core systems related to business approval process are centralized and deployed in HQ operation center; branches access them via enterprise VPN. Standardized processes and systems help XYZ Insurance speed up their business. Beyond utilizing the process systems, the operations center generates companywide statistics report monthly, so that company and branch performance can be monitored.

Here is a sample organizational graph of XYZ Insurance. Organizational units are in blue color, while employees are in orange color. Organization depth varies in different areas, resulting in an unbalanced organization tree.

Figure 2. Organizational chart of XYZ Insurance



There are 3 important processes within XYZ Insurance: IssuePolicy, Claim, and Surrender. In XYZ Insurance's daily business, an employee in HQ, branch or sub-branch launches

one process, an approver from HQ approves, and then the employee who launched the process does the final action.

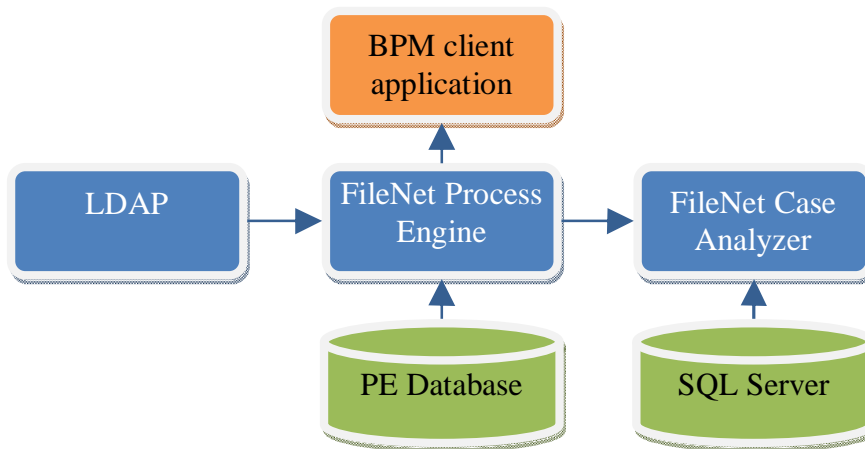
The monthly reports XYZ Insurance needs are as follows:

- Monthly ongoing/completed tasks by employee;
- Monthly ongoing/completed tasks by branch/sub-branch;
- Monthly companywide ongoing/completed processes by process type;
- For each report, growth rate comparing last month and the same month last year;
- Monthly average processing time of each step in a process;
- Monthly 72-hour/7-day/30-day/3-month completion rate for each employee;
- Yearly summary of all above monthly reports.

Solution Architecture

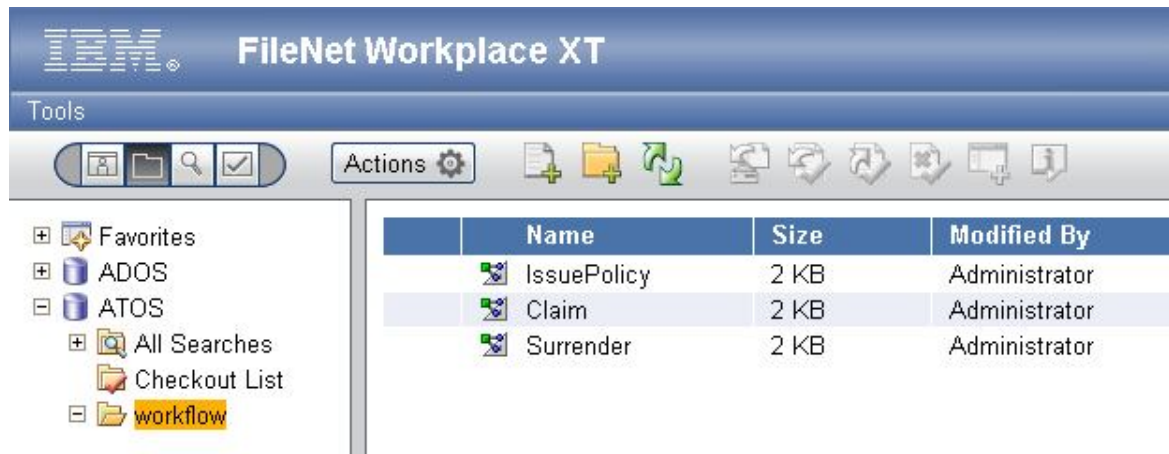
XYZ Insurance developed and deployed an insurance BPM system based on IBM FileNet platforms, which has the following architecture.

Figure 3. A brief architecture diagram of XYZ Insurance BPM System



The organization information is stored in Microsoft Active Directory.

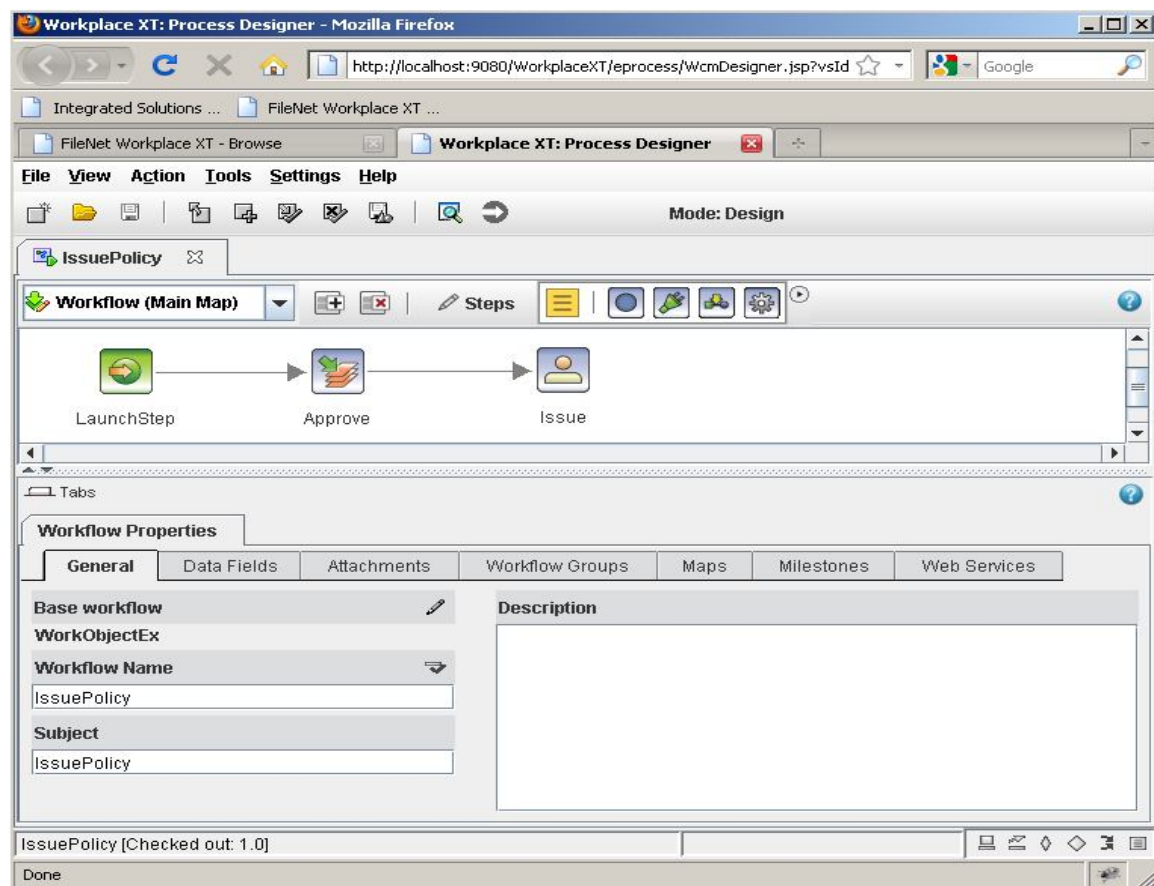
These 3 processes are implemented with FileNet BPM. FileNet BPM is connected to the same Active Directory. Below are the process definitions.

Figure 4. All 3 important processes of XYZ Insurance


The screenshot shows the IBM FileNet Workplace XT interface. On the left is a navigation pane with a tree view containing 'Favorites', 'ADOS', 'ATOS', 'All Searches', 'Checkout List', and 'workflow'. The 'workflow' folder is highlighted. The main area displays a table of processes.

	Name	Size	Modified By
	IssuePolicy	2 KB	Administrator
	Claim	2 KB	Administrator
	Surrender	2 KB	Administrator

All three processes are simplified for this article. Each process contains 3 steps: LaunchStep, Approve and Issue/Claim/Surrender. The Approve steps are handled by the work queue ApproverQueue, the Issue/Claim/Surrender steps are handled by the user who launch the process, saying F_Originator. No data fields are added.

Figure 5. Simplified process definition for IssuePolicy process

All processes share the same DefaultEventLog.

Gap between CA product and requirement

Looking into XYZ Insurance's report requirements some can be met using cubes and reports provided by Case Analyzer, but some cannot.

Table 1. XYZ Insurance's report requirements comparing CA capabilities

Report	CA built-in?	Comment
Monthly ongoing/completed tasks by employee	Yes	Use the "Queue Load" cube
Monthly ongoing/completed tasks by branch/sub-branch	No	CA only provides a "flat" user dimension.
Monthly companywide ongoing/completed processes by process type	Yes	Use the "Workflow count" cube
Growth rate comparing last month and the same month last year in above reports	No	Seems meaningful but no such measure in CA
Monthly average processing time of each step in a process	Yes	Use the "Monthly Avg Step Processing Time" cube
Monthly 72-hour/7-day/30-day/3-month completion rate of each employee	No	Seems meaningful but no such measure in CA
Yearly summary of all above monthly reports	-	Based on above reports

Fortunately, OLAP is an open technology and possible to customize.

This article shows how to produce these missing reports by customizing the CA cubes.

Sample environment setup

The article uses an all-in-one environment for simplicity. This all-in-one environment is based on a Microsoft Windows Server 2003 R2 Enterprise Edition, it covers the following mid-wares:

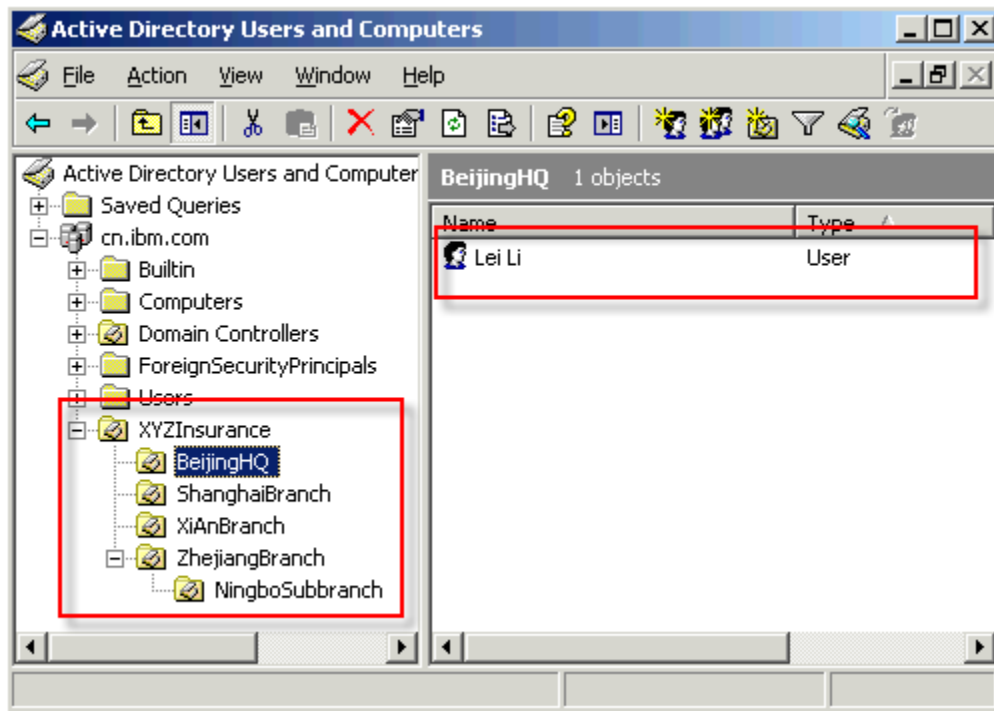
- IBM DB2 9.7
- IBM WebSphere Application Server v7.0
- IBM FileNet CM 5.0
- IBM FileNet PE 5.0
- IBM FileNet Workplace XT
- IBM FileNet Case Analyzer
- Microsoft Active Directory

- Microsoft SQL Server 2005 Enterprise Edition

Above mid-wares are properly installed and configured. FileNet CM and PE are installed on top of DB2, while Case Analyzer is installed on SQL Server.

In Active Directory console, add organizational units and corresponding users listed in [Figure 2](#). Make sure those accounts can be used to login to FileNet.

Figure 6. Organizational units and users in XYZ Insurance



Download the attached XYZInsurance_processes.zip archive file and extract 3 PEP process definitions from it. Bring up Workplace XT, login as administrator, and create a work queue named ApproverQueue in Process Configuration Console. Then transfer the PEP definitions to PE.

Login Workplace XT using different accounts of XYZ Insurance, randomly launch or complete the 3 processes. Once accumulated enough process data, manually turn the CA on to see whether cubes are processed correctly.

Prepare the LDAP user-organization importer

In the part 2 of the series, there will be a need to import user-organization information from LDAP to manually created dimension tables. This section helps prepare the importer application. If you'd like to skip part 2, skip this section as well.

Companies usually use LDAP to store organization information. As mentioned above, in this article Microsoft Active Directory embedded in Windows Server 2003 R2 Enterprise Edition is used. It doesn't make too much difference to use other vendor's LDAP product, IBM Tivoli Directory Server is also a good candidate.

A customized Java application UserOrgSync is developed to read organization information from LDAP and write into above dimension tables.

This Java application is downloadable in article attachments. It's packed as an Eclipse Java project archive. You may import it into Eclipse 3.7 IDE and check the source code. To compile the source code, you need to collect required libraries listed in /lib/readme.txt by yourself:

- commons-collections-3.2.jar
- commons-lang-2.5.jar
- commons-logging-1.0.4.jar
- commons-pool-1.5.4.jar
- geronimo-jpa_3.0_spec-1.0.jar
- geronimo-jta_1.1_spec-1.1.jar
- log4j-1.2.15.jar
- openjpa-1.2.2.jar
- serp-1.13.1.jar
- spring-ldap-core-1.3.1.RELEASE.jar
- spring-ldap-core-tiger-1.3.1.RELEASE.jar
- spring-ldap-test-1.3.1.RELEASE.jar
- spring.jar
- sqljdbc4.jar

The SQL for previous section can be found in the project:

- /src/database/EX_DMOrg.sql
- /src/database/EX_DMUserOrg.sql
- /src/database/EX_DMUser.sql

After the tables and view created, you may make some adjustments, the LDAP and database connection arguments are specified in Spring configuration file:

- /src/resource/context.xml

Double check the following lines in case you're not using default values:

Listing 1. Spring context configuration XML in UserOrgSync project

```
<bean id="processAnalyzerService" class="com.ibm.cn.ecm.ca.ProcessAnalyzerService">
    .....
    <property name="rootDn" value="ou=XYZInsurance,dc=cn,dc=ibm,dc=com" />
</bean>

<bean id="contextSource"
    class="org.springframework.ldap.core.support.LdapContextSource">
    <property name="url" value="ldap://localhost:389"/>
    <property name="userDn" value="administrator@cn.ibm.com" />
    <property name="password" value="filenet" />
</bean>
```

```
.....  
</bean>  
  
.....  
  
<bean id="dataSource"  
    class="org.springframework.jdbc.datasource.DriverManagerDataSource"  
    lazy-init="true">  
    .....  
    <property name="url" value="jdbc:sqlserver://localhost:1433;" />  
    <property name="username" value="ca_client_user" />  
    <property name="password" value="filenet" />  
</bean>
```

Do not run it until target dimension tables are created in part 2 of the series.

Let's explain a little bit about the source code of the application.

This project leverages open source libraries, including Spring, Spring LDAP, OpenJPA, Apache Commons. Spring provides IoC container, Spring LDAP helps reading LDAP, OpenJPA and Spring JpaTemplate provides ORM and DB access abilities.

The key class of the project is `com.ibm.cn.ecm.ca.ProcessAnalyzerService`, the `importOrgAndUser()` method update organizational unit and user data in extended tables for CA. Internally it does the following operations:

- Retrieve a tree of organizational units and users from LDAP;
- Convert the tree to a new tree of Organization and UserOrganization;
- Clear existing data in CA extended tables;
- Persist the new tree into CA extended tables.

Listing 2. Implementation of `importOrgAndUser()` method

```
public void importOrgAndUser() {  
    OrgUserBuildingVisitor visitor = new OrgUserBuildingVisitor();  
    ldapTreeBuilder.getLdapTree(new DistinguishedName(rootDn)).traverse(visitor);  
    final Organization rootOrg = visitor.getRoot();  
  
    getJpaTemplate().execute(new JpaCallback() {  
        public Object doInJpa(EntityManager em)  
            throws PersistenceException {  
            em.getTransaction().begin();  
  
            Query clearUserOrgQuery = em.createQuery("DELETE FROM UserOrganization u");  
            clearUserOrgQuery.executeUpdate();  
            Query clearOrgQuery = em.createQuery("DELETE FROM Organization o");  
            clearOrgQuery.executeUpdate();  
  
            em.persist(rootOrg);  
  
            em.getTransaction().commit();  
            return null;  
        }  
    });  
}
```

Above method leverages the `OrgUserBuildingVisitor`, which implements the Visitor design pattern. The `OrgUserBuildingVisitor` visits nodes of a LDAP tree and builds up a new tree with entities `Organization` and `UserOrganization`.

The service can be optimized in various aspects. For example, maybe not all users recorded in LDAP participate in the processes; the full organization-user tree will be an overhead. CA records a list of process users, so the service may retrieve it by using the SQL below, and then use it as a user filter when building the tree.

Listing 3. Filtering LDAP users by CA dimension data

```
SELECT UserName FROM D_DMUser WHERE Userid > 0
```

Summary

This part introduces IBM FileNet Case Analyzer and XYZ Insurance's requirements, introduces solution architecture as context, and points out the gap between CA product and requirements. This part also explains how to setup a sample environment for later parts' use.

The part 2 will continue to introduce how to implement an organization-user drill up or drill down in CA cubes.